K.L.N COLLEGE OF INFORMATION TECHNOLOGY, POTTAPALAYAM

DEPARTMENT OF (COMPUTER SCIENCE AND ENGINEERING)

Sub. Code & Sub. Name: HX 8001 & Professional Readiness for Innovation, Employability and Entrepreneurship

Project Report

Fertilizers Recommendation System For Disease Prediction

Team ID: PNT2022TMID52484

Submitted by,

1. KEERTHIKA A (910719104015) – Team Leader

2. SWATI V (910719104029) – Team Member

3. DIVYA T K (910719104010) – Team Member

4. DINESH A J (910719104008) – Team Member

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 PROJECT OVERVIEW OF THE FERTILIZER RECOMMENDATION SYSTEM:

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques. An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

1.2 PURPOSE OF FERTILIER RECOMMENDATION SYSTEM

The purpose of the fertilizer recommendation system is to provide farmers with actionable fertilizer advice. For scalability, the recommendations will be disseminated through information and communication technology channels, primarily mobile phones. Bridging information gaps through communication technology channels can overcome the problem of access to trained personnel, making knowledge more widely available. Thus, an integrated, end-to-end solution in the form a point-of-use sensor combined with a recommendation system is intended to enable widespread adoption of site-specific nutrient management and facilitate the shift towards sustainable agriculture.

CHAPTER-2
LITERATURE SURVEY

2.1 PROBLEM EXISTING

Soils naturally contain many nutrients like nitrogen, phosphorous, calcium, and potassium. These nutrients allow plants to grow. When soil nutrients are missing or in short supply, plants suffer from nutrient deficiency and stop growing. When the nutrient level is too low, the plant cannot function properly and produce the food necessary to feed the worlds' population. Once crops are harvested for human consumption, the natural supply of nutrients in the soil must be "re-filled". This is why farmers add nutrients to their soils. Nutrients can be added from a variety of sources—organic matter, chemical fertilizers, and even by some plants. This maintains the soil fertility, so the farmer can continue to grow nutritious crops and healthy crops.

2.2 REFERENCES

2.2.1 Dr.P.Pandi Selvi, P. Poornima, Soil Based Fertilizer Recommendation System for Crop Disease Prediction System, International Journal of Engineering Trends and Applications (IJETA) – Volume 8 Issue 2,Mar-Apr 2021

These proposed systems were organized in such a way, to analyze the soil type, diseases in the leaves and finally to recommend the appropriate fertilizer to the farmers, that may be of great help to them. Plant disease, especially on leaves, is one of the major factors that reduce the yield in both quality and quantity of the food crops.

2.2.2 S.Yegneshwar Yadhav, T.Senthilkumar, S.Jayanthy, J.Judeson Antony Kovilpillai, Sri Ramakrishna Engineering College, Coimbatore, Plant Disease Detection and Classification using CNN Model with Optimized Activation Function, Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020) IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

This proposed work uses unsupervised learning for extracting features and pattern on its own. CNN uses existing non – linear activation functions for classification. The convolutional Neural Network model was implemented the plant diseases.

2.2.3 Shloka Gupta, Nishit Jain , Akshay Chopade, Aparna Bhonde, Farmer's Assistant: A Machine Learning Based Application for Agricultural Solutions, arXiv:2204.11340v1 [cs.LG] 24 Apr 2022

With this system, they are successfully able to provide several features, fertilizer recommendation using a rule-based classification system, andcrop disease detection using Efficient Net model on leaf images. The user can provide the input using forms on our user interface and quickly get their results.

2.2.4 G. Elizabeth Rani, Ede Venkatesh, Karnam Balaji, Balasaraswathi Yugandher, Adiki Nithin Kumar, SakthiMohan. M , An automated prediction of crop and fertilizer disease using Convolutional Neural Networks (CNN), 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) | 978-1-6654-3789-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICACITE53722.2022.9823767

It is a user-friendly intelligent setup for farmers was built with the best accurate results in CNN module. This module was used to fit the

fertilizer recommendation system which gives us a 99.9% of accuracyand an error rate of 0.1%.

2.3 PROBLEM STATEMENT

As agriculture struggles to support the rapidly growing global population, plant disease reduces the production and quality of food, fiber and biofuel crops. Losses may be catastrophic or chronic, but on average account for 42% of the production of the six most important food crops. Crop losses tend to be greatest in tropical countries where environmental conditions are particularly favorable, incomes are low and knowledge and investments in crop health management are minimal. Disease losses can mean that communities become dependent on imported foods, often replacing a balanced diet with processed foods that create further health problems. This issue is focused on research aimed at improving food security by reducing crop losses, particularly for low-income farmers. Manuscripts are invited that describe research into improving food security by reducing yield losses. Such research may include plant pathology, agronomy, entomology, weed science, farm management, improving resilience to abiotic constraints, postharvest handling, food safety, improved market access, the role of biotechnology, technology transfer, extension, education, policy and any other related topics.

CHAPTER-3
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

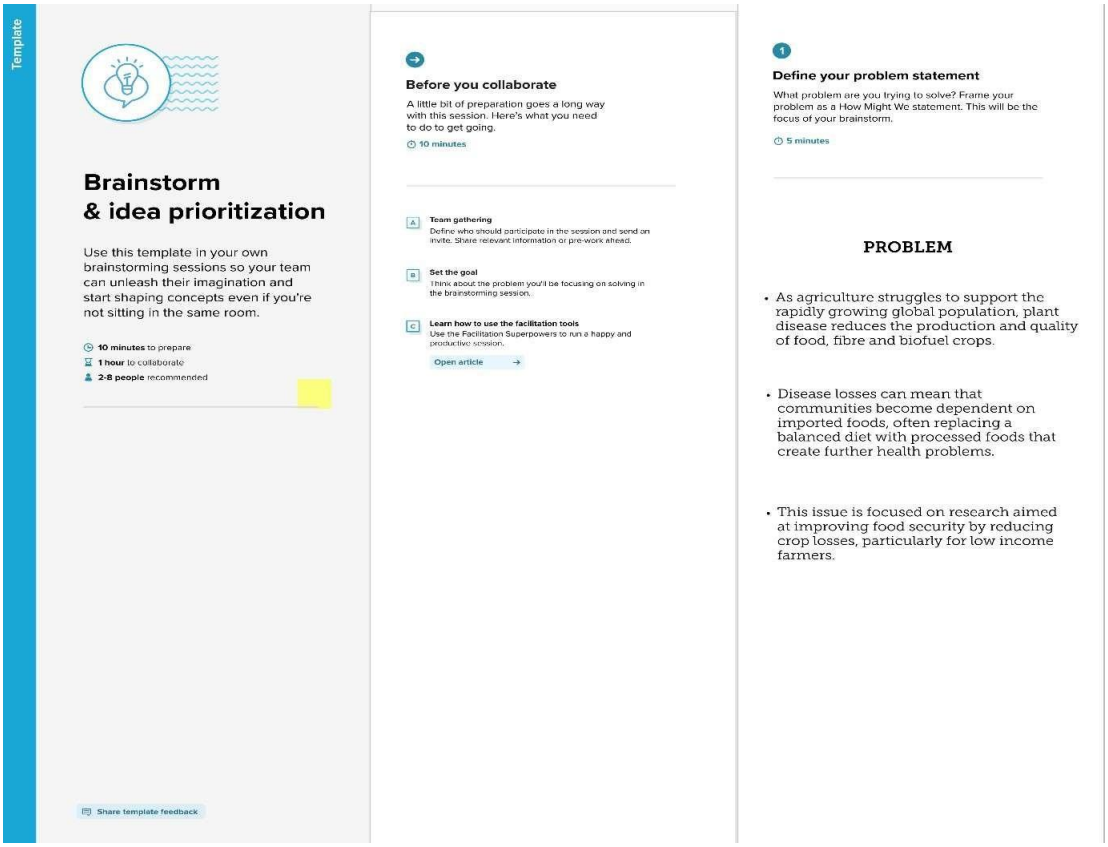
3.1.1 The **Says** quadrant contains what the user says out loud in an interview or some other usability study. Ideally, it contains verbatim and direct quotes from research.

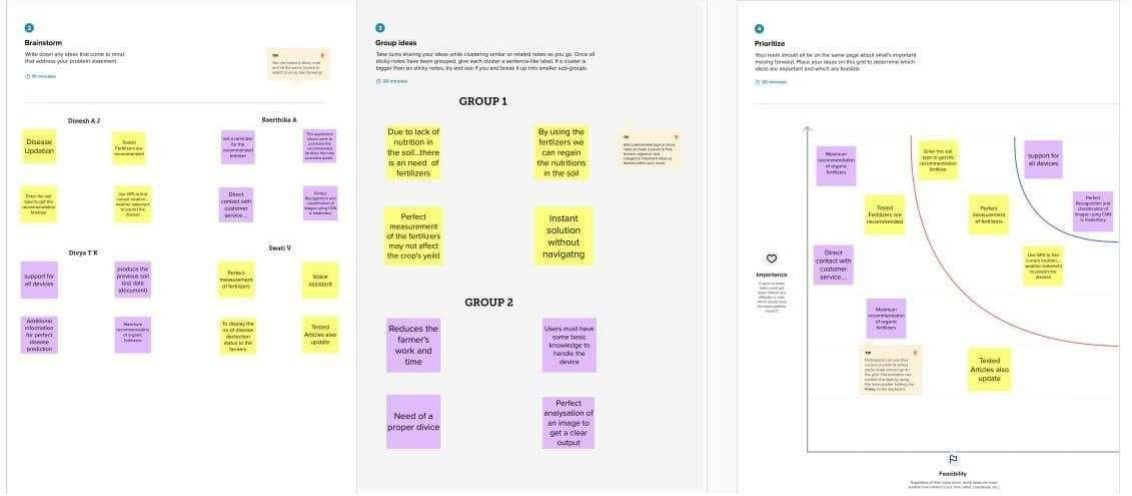
3.1.2 The **Thinks** quadrant captures what the user is thinking throughout the experience. Ask yourself (from the qualitative research gathered): what occupies the user's thoughts? What matters to the user? It is possible tohave the same content in both Says and Thinks. However, pay special attention to what users think, but may not be willing to vocalize. Try to understand why they are reluctant to share — are they unsure, self- conscious, polite, or afraid to tell others something?

3.1.3 The **Does** quadrant encloses the actions the user takes. From the research, what does the user physically do? How does the user go aboutdoing it?

3.1.4 The **Feels** quadrant is the user's emotional state, often represented as an adjective plus a short sentence for context. Ask yourself: what worries the user? What does the user get excited about? How does the user feel about the experience?

3.2 IDEATION & BRAINSTORMING





Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members. In other words, brainstorming is a situation where a group of people meet to generate new ideas and solutions around a specific domain of interest by removing inhibitions. People are able to think more freely and they suggest as many spontaneous new ideas as possible. All the ideas are noted down without criticism and after the brainstorming session the ideas are evaluated.

3.3 PROPOSED SOLUTION

1. Problem Statement (Problem to be solved)

Plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on

plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques. An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

2. Idea / Solution description

An automated system is built that takes the input as picture of leaves which is uploaded by the user, identifies different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the fertilizer needed for the plant.

3. Novelty / Uniqueness

Automatic detection of plant diseases using deep learning models.

4. Social Impact / Customer Satisfaction

- Providing information about the nearby fertilizer store.
- Providing web sites about the availability of fertilizer that is being recommend get remedy for the disease.
- Providing additional information about the plant.

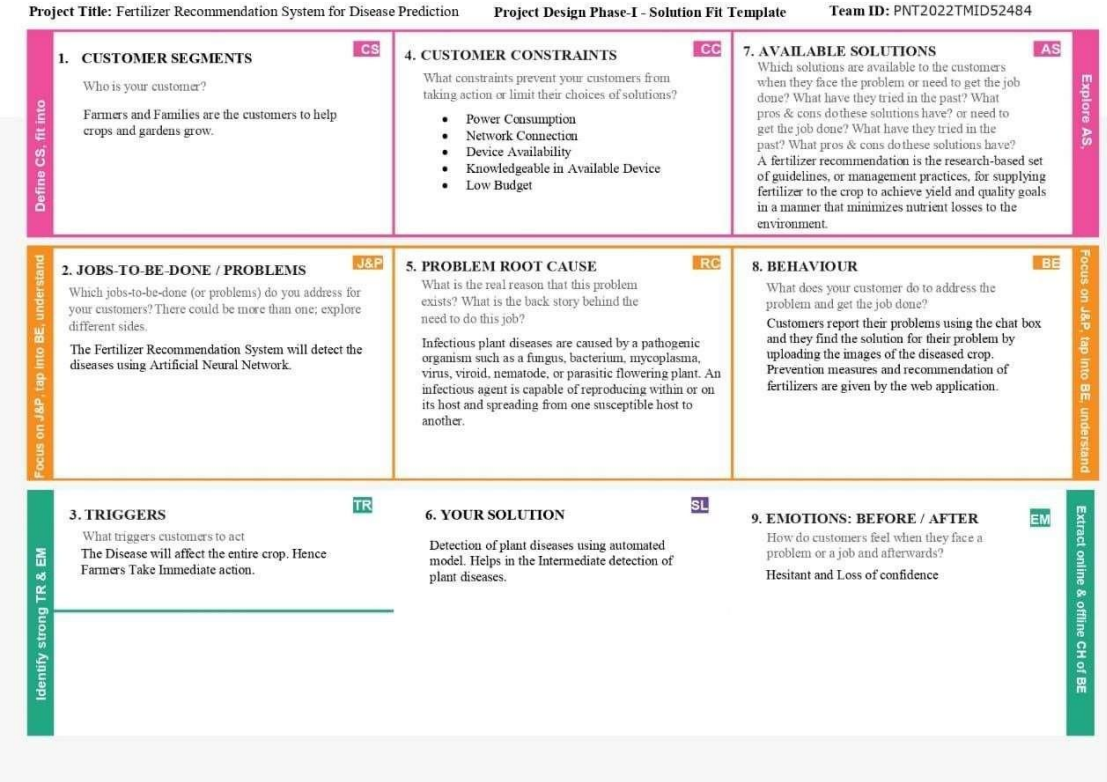
5. Business Model (Revenue Model)

- Collaboration with fertilizer vendors.
- Providing solutions for farmer's problem.
- Increase in crop productivity**.**

6. Scalability of the Solution

For scalability, the recommendations will be disseminated through information and communication technology channels, primarily mobile phones. Bridging information gaps through communication technology channels can overcome the problem of access to trained personnel, making knowledge more widely available.

3.4 PROBLEM SOLUTION FIT



A problem-solution-fit occurs if a startup has proved both:

1) That there is a 'problem worth solving' for one or more clearly defined customer groups

2) That there is evidence that these customer groups would consider the value proposition of the solution the firm proposes.

Having reached problem-solution fit means that you've understood your customers' needs/ jobs-to-be-done and their desired outcomes. It also means that you've created a solution that addressed these needs in a unique way, as customer feedback suggests so (validation of desirability). However, contrary to widely-held opinion it does not mean that you already have evidence that they would really buy it. This next inflection point in lean validation is called product-market fit.

CHAPTER-4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

1. User Registration - Registration through website or application. Registration through social media like insta, facebook,whatsapp.

2. User Confirmation - Verify via mail, verify via OTP through SMS

3. User Login - Login through website or app using their Username and Password

4. User Access - Allows the app requirements

5. User Guide - Guides the basic steps for using the application

6. User Upload - User should be able to send the data

7. User solution - Data report should be generated and delivered to user in every24 hours

8. User Data sync - API interface to invoice system

4.2 NON-FUNCTIONAL REQUIREMENTS

1. Usability - User can easily learn and use the application and they can send their queries and data through the application. Usability can be assessed from different points of view:

- Efficiency of use
- Low perceived workload
- Intuitiveness

2. Security - Security requirements ensure that the software is protected from unauthorized access to the system and its stored data.

3. Reliability - Reliability defines how likely it is for the software to work without failure for a given period of time.

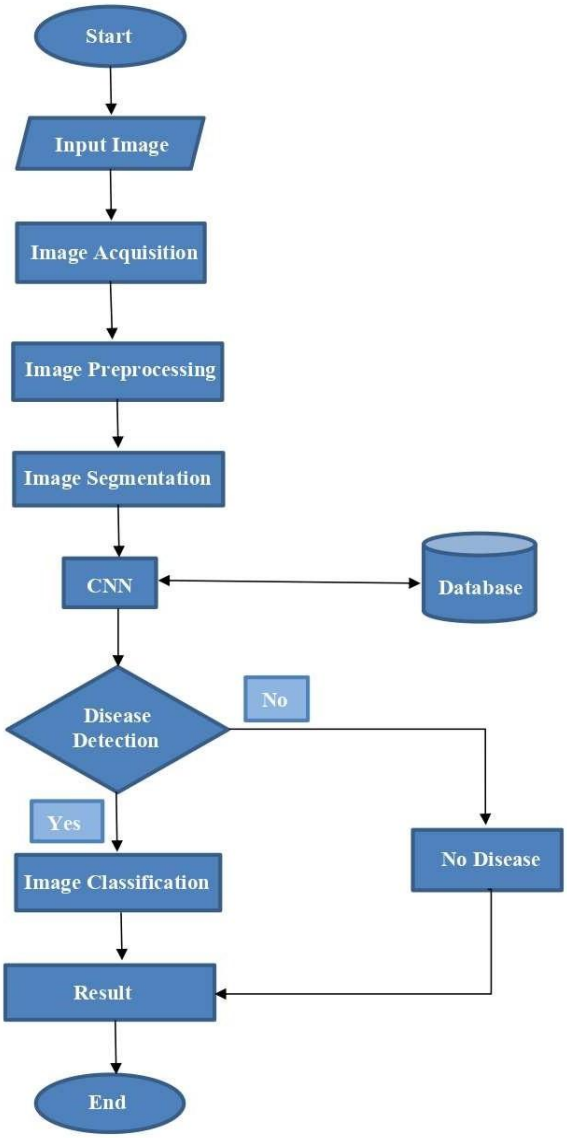
4. Performance - Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it.

5. Availability - Availability is gauged by the period of time that the system's functionality and services are available for use with all operations.

6. Scalability - Scalability requirements describe how the system must grow without negative influence on its performance.

CHAPTER-5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



This is a graphical representation of the flow of data about how the entire prediction and recommendation process is carried out. This data flow diagram undergoes the following steps,

1. **Image Input**

The user must browse and upload the defected leaf image.

2. **Image Acquisition**

Image acquisition is the action of retrieving an image from a source, usually hardware systems like cameras, sensors, etc. It is the first and the most important step in the workflow sequence because, without an image, no actual processing is possible by the system. The image that is acquired by the system is usually completely unprocessed.

3. **Image Preprocessing**

Image preprocessing is the steps taken to format images before they are used by model training and inference. This includes, but is not limitedto, resizing, orienting, and color corrections .Image preprocessing may also decrease model training time and increase model inference speed. If input images are particularly large, reducing the size of these images will dramatically improve model training time without significantly reducing model performance.

4. **Image Segmentation**

It is the process of dividing an image into different regions based on the characteristics of pixels to identify objects or boundaries to simplify an image and more efficiently analyze it.

5. **CNN Model**

The next process is CNN model. This is used for identifying and recognizing objects. So, this makes them highly suitable for computer vision. CNN uncover key information in both time series and image data. Thus, it is highly valuable for image related tasks. This model undergoes with the database. Here, the database is the images that are retrieved from the several source.

6. Disease Deduction

After undergone through the CNN process, the flow diagram reaches the decision-making area called disease deduction. If the process deducted the presence of disease, then it goes to the further step otherwise, it gives the result in such a way that there is no disease is deducted.

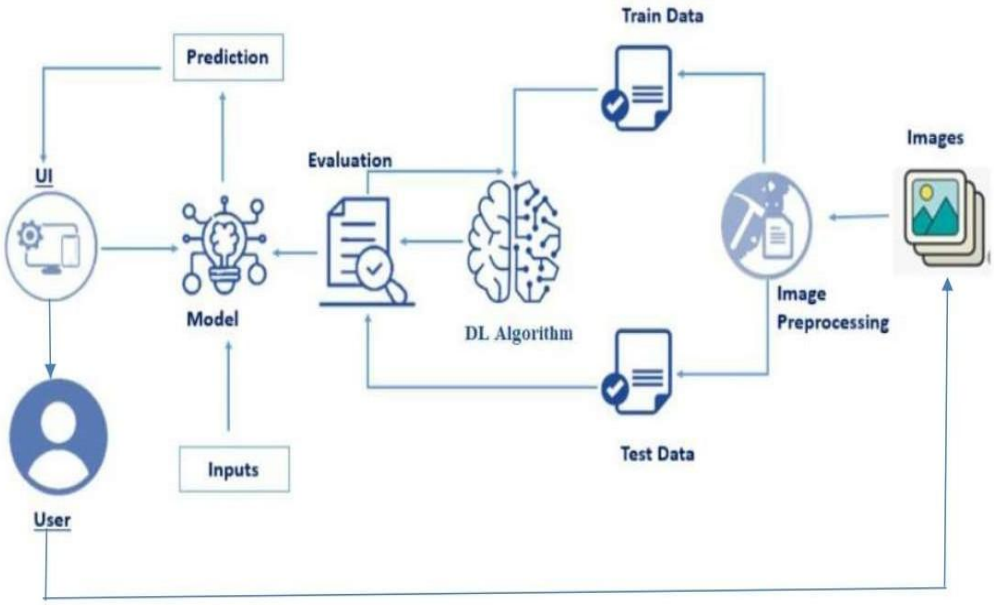
7. Image Classification

When the disease is predicted**,** the image will further undergo about to which disease the plant is affected and to which family does the leaf belongs to whether it is a fruit or a vegetable.

8. Result

As a result, the user may get to know about the correct disease that is predicted and the recommended fertilizers.

5.2 SOLUTION & TECHNICAL ARCHITECTURE



This architecture will explain about the technical process that happened in the system. First, the user gives the input image. So, the image undergoes the process of image preprocessing then the given data is trained and tested. The trained data is computed by the DL algorithm. The data is evaluated and the remaining process is carried out the CNN model. Here, the input is called as data if the data is predicted then output is displayed on the User Interface (UI). The technical process is explained detailed below,

1. Image Preprocessing

Pre-processing is a common name for operations with images at the lowest level of abstraction — both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (bright nesses). The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g., rotation, scaling, translation) are classified among pre-processing methods here since similar techniques are used.

Image pre-processing methods are classified into four categories according to the size of the pixel neighborhood that is used for the calculation of a new pixel brightness that deals with pixel brightness transformations, describes geometric transformations, considers preprocessing methods that use a local neighborhood of the processed pixel and briefly characterizes image restoration that requires knowledge about the entire image.

2. Deep Learning Algorithm

Deep learning is a subset of machine_learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing

it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labeled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn’t require labeled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics.

3. CNN Model

A Convolutional Neural Network is a special class of neural networks that are built with the ability to extract unique features from image data. For instance, they are used in face detection and recognition because they can identify complex features in image data. Like other types of neural networks, CNNs consume numerical data. Therefore, the images fed to these networks must be converted to a numerical representation. Since images are made up of pixels, they are converted into a numerical form that is passed to the CNN. However, as we will discuss in the upcoming section, the entire numerical representation is not passed into the network. The steps involved in building and training a CNN model are,

3.1 Convolution

Reducing the size of the numerical representation sent to the CNN is done via the convolution operation. The result of the convolution operation is referred to as a feature map, convolved feature, or activation map. Applying a feature detector is what leads to a feature map. The feature detector is also known by other names such as kernel or filter. The kernel is usually a 3 by 3 matrix. Performing an element-wise multiplication of the kernel with the input image and summing the values, outputs the feature map. This is done by sliding the kernel on the input image. The sliding happens in steps known as strides. The strides and the size of the kernel can be set manually when creating the CNN.

3.2 Padding

Padding involves increasing the size of the input image by "padding" the images with zeros. As a result, applying the filter to the image leads to a feature map of the same size as the input image. Padding reduces the amount of information lost in the convolution operation. It also ensures that the edges of the images are factored more often in the convolution operation.

3.3 Activation Functions

A Rectified Linear Unit (ReLU) transformation is applied after every convolution operation to ensure non-linearity. ReLU is the most popular activation function but there are other activation functions to choose from. After the transformation, all values below zero are returned as zero while the other values are returned as they are.

3.4 Pooling

In this operation, the size of the feature map is reduced further. There are various pooling methods. Pooling forces the network to identify key features in the image irrespective of their location. The reduced image size also makes training the network faster.

3.5 Dropout Regularization

Applying Dropout Regularization is a common practice in CNNs. This involves randomly dropping some nodes in layers so that they are not updated during back-propagation. This prevents overfitting.

3.5 Flattening

Flattening involves transforming the pooled feature map into a single column that is passed to the fully connected layer. This is a common practice during the transition from convolutional layers to fully connected layers.

3.6 Fully connected layers

The flattened feature map is then passed to a fully connected layer. There might be several fully connected layers depending on the problem and the network. The last fully connected layer is responsible for outputting the prediction. An activation function is used in the final layer depending on the type of problem.

4. User Interface

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

5.3 USER STORIES

This describes which type of users do a particular task under the acceptance criteria. The user stories explain the priority of the task and the task comes under which print of release.

User type	Functional requirements (epic)	User story number	User story/ task	Acceptance criteria	Priority	Release
Admin	Collection of data	USN-1	Collect the data from various resources and analyze the collected data.	I can collect the dataset	Low	Sprint-1
Admin	Data preprocessing	USN-2	Importing the required libraries then loading the data and reshaping as a preprocessed data for making them as a same format.	I can import the libraries for preprocessing	Low	Sprint-1

Admin	Model creation and Training (Fruits)	USN-3	Create a model which can classify fruit plants from given images. We also need to train and test the model.	I can create a model for fruit disease	High	Sprint-2
Admin	Model creation and training (Vegetables)	USN-4	Create a model which can classify vegetables plants from given images. We also need to train and test the model.	I can create a model for vegetable disease	High	Sprint-2
User	Registration	USN-5	As a user, I can register for the application by entering my email, password, and	I can register and access the portal	Medium	Sprint-3

			confirming my password			
User	Login	USN-6	As a user, I can log into the application by entering email & password	I can login to enter into the home page	Medium	Sprint-3
User	Upload page	USN-7	As a user, I will be redirected to a page where I can upload my pictures of crops	I can upload my defected leaf image	High	Sprint-3
User	Suggestion results	USN-8	As a user, I can view the results and then obtain the suggestions provided by the ML	I can see the predicted disease and recommended fertilizers to be used	High	Sprint-3

			model			
Admin	Flask	USN-9	Deployment of all codes in our local system using the framework called flask	I can use the python code using the framework called flask for backend	High	Sprint-4

CHAPTER-6

PROJECT PLANNING &SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
1	Collection of data	USN-1	Collect the data from various resources and analyze the collected data.	2	Low	Keerthika Swati Divya Dinesh
1	Data preprocessin	USN-2	Importing the required libraries then	3	Low	Keerthika

	g		loading the data and reshaping as a preprocessed data for making them as a same format.			Swati Divya Dinesh
2	Model creation and Training (Fruits)	USN-3	Create a model which can classify fruit plants from given images. We also need to train and test the model.	8	High	Keerthika Swati Divya Dinesh
2	Model creation and training (Vegetables)	USN-4	Create a model which can classify vegetables plants from given images. We also need to train and test the model	8	High	Keerthika Swati Divya Dinesh

3	Registration	USN-5	As a user, I can register for the application by entering my email, password, and confirming my password	4	Medium	Keerthika Swati Divya Dinesh
3	Login	USN-6	As a user, I can log into the application by entering email & password	4	Medium	Keerthika Swati Divya Dinesh
3	Upload page	USN-7	As a user, I will be redirected to a page where I can upload my pictures of crops	4	High	Keerthika Swati Divya Dinesh
3	Suggestion results	USN-8	As a user, I can view the results and then obtain	5	High	Keerthika Swati

			the suggestions provided by the ML model			Divya Dinesh
4	Flask	USN-9	Deployment of all codes in our local system using the framework called flask	8	High	Keerthika Swati Divya Dinesh

6.2 SPRINT DELIVERY SCHEDULE

This is a preplanned schedule for our sprint delivery. This shows our exact progress and path for completing our project.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	17	6 Days	24 Oct 2022	29 Oct 2022	17	30 Oct 2022

Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	12	06 Nov 2022
Sprint-3	19	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	15	6 Days	14 Nov 2022	19 Nov 2022	14	20 Nov 2022

6.3 REPORTS FROM JIRA

Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.

To practice agile methodologies, Jira Software provides scrum and Kanban boards out-of-the-box. Boards are task management hubs, where tasks are mapped to customizable workflows. Boards provide transparency across teamwork and visibility into the status of every work item. Time tracking capabilities and real-time performance reports (burn-up/down charts, sprint reports, velocity charts) enable teams to closely monitor their productivity over time. We have designed our sprint using scrum board.

6.3.1 Scrum Board

Scrum boards are visual project management tools that help Scrum teams visualize backlog items and work progress. Scrums are broken down into time-

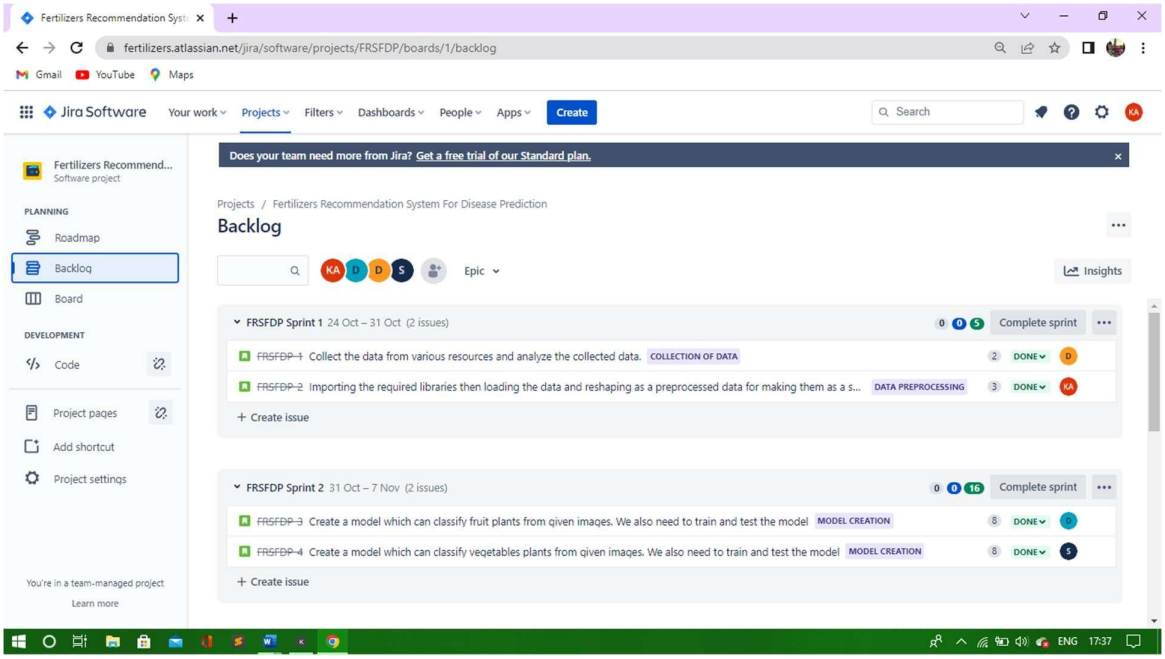
boxed iterations named sprints, lasting between one and four weeks. Most development teams shoot for two-week sprints. Scrum boards are visual project management tools that help Scrum teams visualize backlog items and work progress. Scrum boards track individual sprints and help team members visualize their progress. Scrum boards are also known as Scrum task boards or sprint boards. Humans are visual creatures, and the Scrum board builds on that by helping team members better understand the sprint's progress.

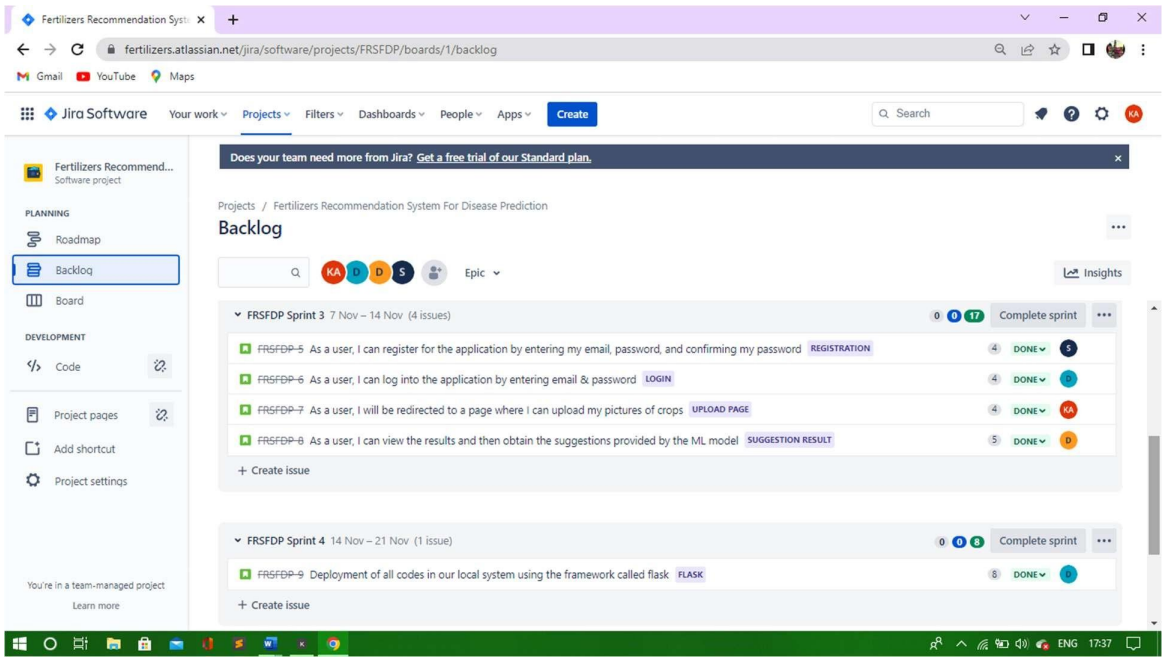
Functions of Scrum Board

- Increase communication and transparency
- Promote sprint planning and iterative development

Scrum Backlog

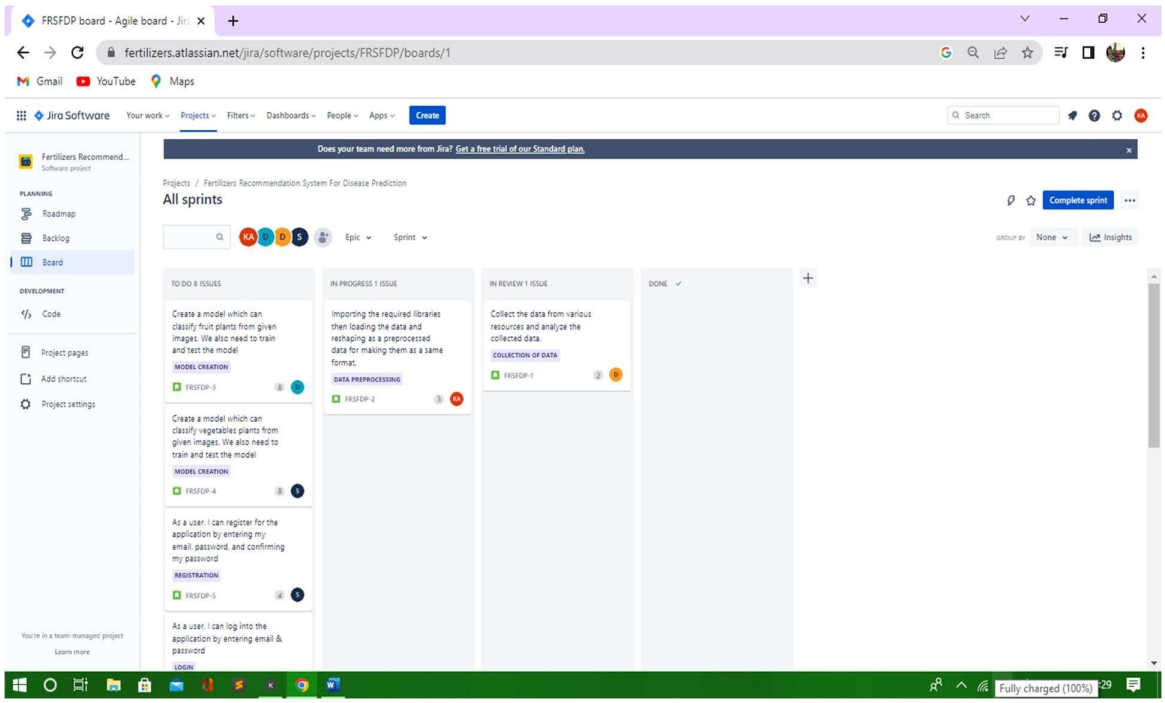
Here, we have given our sprints titles and all our tasks are imported in the backlog.

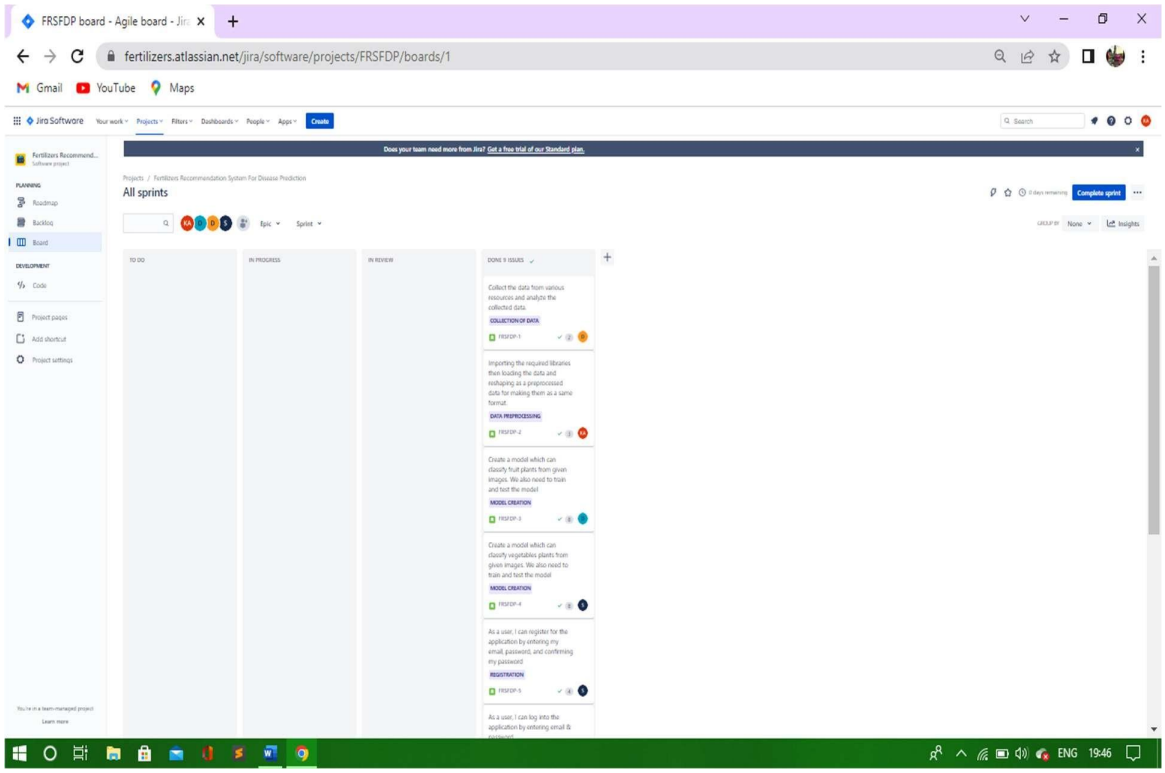




Scrum Board

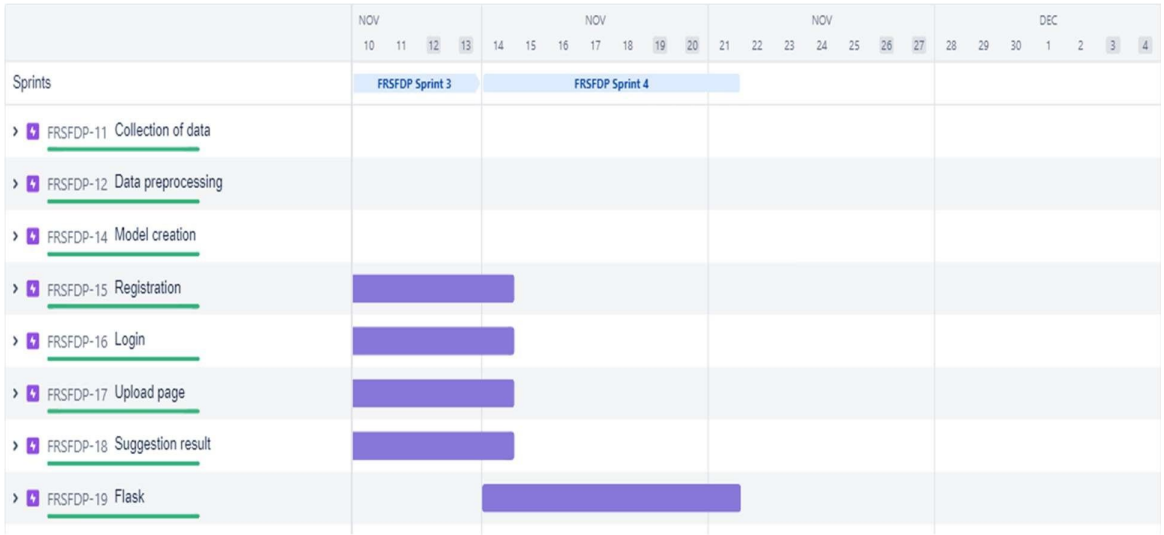
This board will show the status of our sprints. If we completed the assigned level the owner can drag the sprint to the next level.





Road Map

This is the overall journey of the sprints, which shows the timelines of the progress.



CHAPTER-7

CODING & SOLUTIONING

7.1 FEATURE-1

Our project front-end is done using HTML5, CSS and Java Script.

7.1.1 HTML (Hyper Text Markup Language)

HTML stands for Hypertext Markup Language. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a webpage) such as tags and attributes. HTML is the default language of websites and web-based documents. It helps a browser understand the structure and style of a document or files for viewing over the internet. It allows your web pages to host audio, videos, spreadsheets, and other applications. It also facilitates navigation within web pages or between websites through hypertext. Moreover, website makers can use HTML to design forms for ordering products, making reservations, or searching for information. HTML5 supports new kinds of form controls. HTML5 also introduced several semantic tags that clearly describe the content, such as <article>, <header>, and <footer>.

HTML Code

Home Page

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
```

```
href='https://fonts.googleapis.com/css?family=Open+Sans+Condense
d:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin
Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header
{ top:0;
margin:0px
left: 0px;
right: 0px;
position: fixed;
background-color: #28272c;
color: white;
box-shadow: 0px 8px 4px grey;
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav
  { overflow:
  hidden;
  background-color: #333;
}
.topnav-right a
{ float:   left;
color: #f2f2f2; text-
align: center;padding:
14px 16px;
  text-decoration: none;
  font-size: 18px;
}
.topnav-right a:hover
  { background-color:
  #ddd;color: black;
}
.topnav-right a.active
  { background-color:
  #565961;color: white;
}
.topnav-right
  {float: right;
```

```
    padding-right:100px;
  }
  body {
    font-family:'Times New Roman', Times, serif;
    background-image: url("..static/images/s1.jpg");
    background-color:#ffffff;
    background-repeat:  no-repeat;
    background-size:cover;
    background-position:  0px  0px;
    }
    .button {
    background-color:  #28272c;
    border: none;
    color:   white;
    padding: 15px 32px;
    text-align: center;
    text-decoration:  none;
    display:  inline-block;
    font-size: 16px;
    border-radius: 12px;
  }
  .button:hover {
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0
  rgba(0,0,0,0.19);
  }
  form {border: 3px solid #f1f1f1; margin-left:400px;margin-
  right:400px;}
  input[type=text], input[type=password]
    {width: 100%;
    padding: 12px 20px;
    display: inline-block;
    margin-bottom:18px;
    border: 1px solid #ccc;
    box-sizing:  border-box;
  }
  button {
    background-color:  #28272c;
    color: white;
    padding: 14px 20px;
    margin-bottom:8px;
    border: none;
    cursor: pointer;
    width: 15%;
    border-radius:4px;
  }
  button:hover
    { opacity: 0.8;
  }
  .cancelbtn
    { width:
    auto;
padding:10p,18px;
```

```
 background-color: #f44336;
}
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}
img.avatar
  { width: 30%;
  border-radius: 50%;
}
.container
  { padding: 16px;
}
span.psw
  { float:
  right;
  padding-top: 16px;
}
/* Change styles for span and cancel button on extra small
screens */
@media screen and (max-width: 300px)
  {span.psw {
     display: block;
     float: none;
  }
  .cancelbtn
     { width: 100%;
  }
}
.home{ margin:
80px;
  width:   84%;
  height: 500px;
  padding-top:10px;
  padding-left: 30px;
  }
.login{ margin:
80px;
box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid blue;
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
border: 10px solid blue;}
.mySlides {display:  none;}
img {vertical-alignmiddle;}/*
```

```
.slideshow-container
{max-width: 1000px;
position: relative;
margin: auto;
}
/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom:   8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
.active {
  background-color: #717171;
}
/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}
@-webkit-keyframes fade
{from {opacity: .4}
  to {opacity: 1}
}
@keyframes fade
  { from
  {opacity: .4}
  to {opacity: 1}
}
/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
</style>
</head>
<body>
<div class="header">
```

```
 <div  style="width:50%;float:left;font-size:2vw;text-
align:left;color:white;  padding-top:1%">Plant  Disease
Prediction</div>
  <div  class="topnav-right"style="padding-top:0.5%;">
    <a  class="active"  href="{{  url_for('home')}}">Home</a>
    <a  href="{{  url_for('prediction')}}">Predict</a>
  </div>
</div>
<div  style="background-color:#ffffff;">
<div  style="width:60%;float:left;">
<div style="font-size:40px;color:#013220;font-
family:Montserrat;padding-left:20px;text-align:center;padding-
top:10%;">
<b>Fertilizers  Recommendation  System<br>  For  Disease
Prediction!!</b>
</div><br>
<div style="font-size:20px;color:#ffffff;font-family:Arial
Black;padding-left:70px;padding-right:30px;text-
align:justify;">Agriculture is one of the major sectors worls
wide. Over the years it has developed and the use of new
technologies and equipment replaced almost all the traditional
methods of farming. The plant diseases effect the production.
Identification of diseases and taking necessary precautions is
all done through naked eye, which requires labour and
laboratries. This application helps farmers in detecting the
diseases by observing the spots on the leaves, which inturn
saves effort and labour costs.</div><br><br>
</div>
</div>
<div  style="width:40%;float:right;"><br><br>
</div>
</div>
<div class="home">
<br>
</div>
<script>
var slideIndex = 0;
showSlides();
function showSlides()
  {var i;
  var  slides  =  document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++)
    {slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className  =  dots[i].className.replace("  active",
"");
  }
  slides[slideIndex-1].style.display  =  "block";
```

```
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

Predict Page

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.
css" rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"
></script>
    <script
src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></scrip
t>
    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js
"></script>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condense
d:300' rel='stylesheet' type='text/css'>
```

```
<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin
Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<link href="{{ url_for('static', filename='css/final.css') }}"
rel="stylesheet">
<style>
.header
{ top:0;
margin:0px;
left: 0px;
right: 0px;
position: fixed;
background-color: #28272c;
color: white;
box-shadow: 0px 8px 4px grey;
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav
  { overflow:
  hidden;
background-color: #333;}
.topnav-right
 a { float:   left;
color: #f2f2f2; text-
align: center;
```

```
   padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}
.topnav-right a:hover
  { background-color:
  #ddd;color: black;
}
.topnav-right a.active
  { background-color:
  #565961;color: white;
}
.topnav-right
  {float: right;
  padding-right:100px;
}
.login{
margin-top:-70px;
}
body {
 background-image: url("../static/images/s2.jpg");
  background-color:#ffffff;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
.login{
margin-top:100px;
}
.container{ margin
  -top:40px;
  padding: 16px;
}
select
{ width: 100%;
```

```
margin-bottom: 10px;
background:  rgba(255,255,255,255);border:
none;
outline: none;
padding: 10px;
font-size: 13px;
color: #000000;
text-shadow: 1px 1px 1px rgba(0,0,0,0.3);border:
1px solid rgba(0,0,0,0.3);
border-radius: 4px;
box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
-webkit-transition: box-shadow .5s ease;
-moz-transition: box-shadow .5s ease;
-o-transition: box-shadow .5s ease;
-ms-transition: box-shadow .5s ease;
transition: box-shadow .5s ease;
}
</style>
</head>
<body  style="font-family:Montserrat;overflow:scroll;">
<div class="header">
 <div  style="width:50%;float:left;font-size:2vw;text-
align:left;color:white;  padding-top:1%">Plant  Disease
Prediction</div>
  <div class="topnav-right"  style="padding-top:0.5%;">
 </div>
</div>
<div class="container">
        <div id="content"  style="margin-top:2em">
<div class="container">
 <div class="row">
<div class="col-sm-6 bd" >
<br>
</div>
```

```
<div class="col-sm-6">
<div>
<h4>Drop in the image to get the prediction  </h4>
<form action = "/precations/precations_veg" id="upload-file"
method="post" enctype="multipart/form-data">
<select name="plant">
<option value="select" selected>Select plant type</option>

<option value="fruit">Fruit</option>

<option value="vegetable">Vegetable</option>
</select><br>
<label for="imageUpload" class="upload-label" style="background:
#28272c;">
Choose...
</label>
<input type="file" name="image" id="imageUpload" accept=".png,
.jpg, .jpeg">
</form>
<div class="image-section" style="display:none;">
<div class="img-preview">
<div id="imagePreview">
</div>
</div>
<div>
<button type="button" class="btn btn-info btn-lg " id="btn-
predict" style="background: #28272c;">Predict!</button>
</div>
</div>
<div class="loader" style="display:none;"></div>
<h3>
<span id="result" style="font-size:17px;color:white;font-
family:Arial Black;"> </span>
</h3>
</div>
```

```
</div>
</div>
</div>
</div>
</div>
</body>
<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>
</html>
```

7.1.2 CSS (Cascading Style Sheet)

It is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. It is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, and enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file. The name Cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element.

A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block. In CSS, selectors declare which part of the markup a style applies to by matching tags and attributes in the markup itself. Elements are specified by attributes, in particularity or class. Each web browser uses a layout engine to render webpage, and support for CSS functionality is not consistent between them.

CSS Code

```
.img-preview
    { width: 256px;
    height: 256px;
    position:  relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}
.img-preview>div
    {width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat:  no-repeat;
    background-position: center;
}
input[type="file"]
    {display: none;
}
.upload-label{
    display:  inline-block;
    padding: 12px 30px;
    background: #28272c;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}
.upload
label:hover{ background:
#C2C5A8;color: #39D2B4;
}
.loader {
```

```
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #28272c; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

7.1.3 Java Script

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

Script Code

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();
    // Upload  Preview
    function readURL(input) {
```

```
        if (input.files && input.files[0])
            {var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image',
'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
        readURL(this);
    });
     // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);
        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
```

```
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text('Prediction:  '+data);
                console.log('Success!');},
        });
    });
});
```

7.2 FEATURE 2

The backend of our project was done on python programming language. The website is integrated using the framework called flask. The coding sessions where done with Jupiter notebook.

7.2.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

To build the deep learning models we need the following packages,

7.2.2 Tensor Flow

TensorFlow is an open-source library developed by Google primarily for deep learning applications. It also supports traditional machine learning. TensorFlow was originally developed for large numerical computations without

keeping deep learning in mind. It proved to be very useful for deep learning development. TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors. Multi-dimensional arrays are very handy in handling large amounts of data.

7.2.3 Keras

Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features:

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.
- It supports multiple platforms and backends.
- It is user-friendly framework that runs on both CPU and GPU.
- Highly scalability of computation.

7.2.4 Flask

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are:

- There is a built-in development server and a fast debugger provided.
- Lightweight
- Secure cookies are supported.
- Support for unit testing is built-in.

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect,
url_for
```

```
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session
app = Flask(__name__)
global sess
global graph
graph=tf.compat.v1.get_default_graph()
model = load_model(r"C:\Users\Admin\Desktop\New
folder\vegetables.h5")
model1=load_model(r"C:\Users\Admin\Desktop\New folder\fruit.h5")
@app.route('/')
def home():
    return  render_template('home.html')
@app.route('/prediction')
def prediction():
    return  render_template('predict.html')
@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        f = request.files['image']
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(128, 128))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        plant=request.form['plant']
        print(plant)
         if(plant=="vegetable"):
            preds = model.predict(x)
            preds = np.argmax(preds)
            print(preds)
            df=pd.read_excel('precautions_veg.xlsx')
```

```
            print(df.iloc[preds]['caution']
        else:
            preds = model1.predict(x)
            preds = np.argmax(preds)
           df=pd.read_excel('precautions_fruits.xlsx')
            print(df.iloc[preds]['caution'])
            return  df.iloc[preds]['caution']
if___name__ == "__main__":
    app.run(debug=True)
```

The features and the process that we are done in our system are explained below in detail,

Data Collection

We have Created Train and Test folders with each folder having subfolders with leaf images of different plant diseases. Two datasets will be used, we have created two models one to detect vegetable leaf diseases like tomato, potato, and pepper plants and the second model would be for fruits diseases like corn,peach, and apple.

Image Preprocessing

Before training the model, we have to preprocessed the images and then feed them on to the model for training. We used Keras ImageDataGenerator class for image preprocessing. mage data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Import ImageDataGenerator Library and Configure it ImageDataGenerator class is used to load the images with different modifications like considering the zoomed image, flipping the image and rescaling the images to range of 0 and 1.

```
From tensorflow.keras.preprocessing.image import
ImageDataGenerator

train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip
=True)
test_datagen = ImageDataGenerator(rescale=1./255,)

x_train =
train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/D
ataset/Dataset Plant Disease/fruit-dataset/fruit-
dataset/train',target_size = (128,128),batch_size = 32,
class_mode = 'categorical')
Found 5448 images belonging to 6 classes.

x_test =
test_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/Da
taset/Dataset Plant Disease/fruit-dataset/fruit-
dataset/test',target_size = (128,128),batch_size = 32,class_mode
= 'categorical')
Found 1696 images belonging to 6 classes.

x_train.class_indices

{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import
Dense,Convolution2D,MaxPooling2D,Flatten
```

*Apply ImageDataGenerator functionality to Train and Test set. We have Specified the path of both the folders in the flow_from_directory method. We are importing the images in 128*128 pixels.*

```
from tensorflow.keras.preprocessing.image import
ImageDataGenerator

train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip
=True)
test_datagen = ImageDataGenerator(rescale=1./255,)
```

```
x_train =
train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/D
ataset/Dataset Plant Disease/fruit-dataset/fruit-
dataset/train',target_size = (128,128),batch_size = 32,
class_mode = 'categorical')
Found 5448 images belonging to 6 classes.

x_test =
test_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/Da
taset/Dataset Plant Disease/fruit-dataset/fruit-
dataset/test',target_size = (128,128),batch_size = 32,class_mode
= 'categorical')
Found 1696 images belonging to 6 classes.

x_train.class_indices

{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import
Dense,Convolution2D,MaxPooling2D,Flatten
```

Model Building for Fruit & Vegetable Disease Prediction

The model building has been done by the following activities,

1. Importing the Libraries

We have imported the libraries that are required to initialize the neural network layer, and created and added different layers to the neural network model.

```
from tensorflow.keras.preprocessing.image import
ImageDataGenerator

train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip
=True)
test_datagen = ImageDataGenerator(rescale=1./255,)

x_train =
train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/D
ataset/Dataset Plant Disease/fruit-dataset/fruit-
```

```
dataset/train',target_size = (128,128),batch_size = 32,
class_mode = 'categorical')
Found 5448 images belonging to 6 classes.

x_test =
test_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/Da
taset/Dataset Plant Disease/fruit-dataset/fruit-
dataset/test',target_size = (128,128),batch_size = 32,class_mode
= 'categorical')
Found 1696 images belonging to 6 classes.

x_train.class_indices

{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import
Dense,Convolution2D,MaxPooling2D,Flatten
```

2. Initializing the Model

The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add () method.

```
model=Sequential()
```

3. Adding CNN Layers

We will be adding three layers for CNN

- Convolution layer
- Pooling layer
- Flattening layer

The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes a number of feature detectors, feature detector size, expected input shape of the

image, and activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image).

Activation Function: These are the functions that help us to decide if we need to activate the node or not. These functions introduce non-linearity in the networks.

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activat
ion='relu'))
```

After the convolution layer, a pooling layer is added. Max pooling layer can be added using MaxPooling2D class. It takes the pool size as a parameter. Efficient size of the pooling matrix is (2,2). It returns the pooled feature maps.

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

The flatten layer is used to convert n-dimensional arrays to 1-dimensional arrays. This 1D array will be given as input to ANN layers.

```
model.add(Flatten())
```

4. Adding Dense Layers

The name suggests that layers are fully connected (dense) by the neurons in a network layer. Each neuron in a layer receives input from all the neurons present in the previous layer. Dense is used to add the layers.

```
model.add(Dense(6,activation='softmax'))
```

5. Train and Save the Model

After adding all the required layers, the model is to be compiled. For this step, loss function, optimizer and metrics for evaluation can be passed as arguments.

```
model.compile(loss='categorical_crossentropy',optimizer='adam',m
etrics=['accuracy'])
```

Fit the neural network model with the train and test set, number of epochs and validation steps. Steps per epoch is determined by number of training images//batch size, for validation steps number of validation images//batch size.

```
model.fit_generator(x_train,steps_per_epoch=len(x_train),validat
ion_data=x_test,validation_steps=len(x_test),epochs=10)
```

The weights are to be saved for future use. The weights are saved in as .h5 file using save().

```
model.save('fruit.h5')
```

similarly for vegetable,

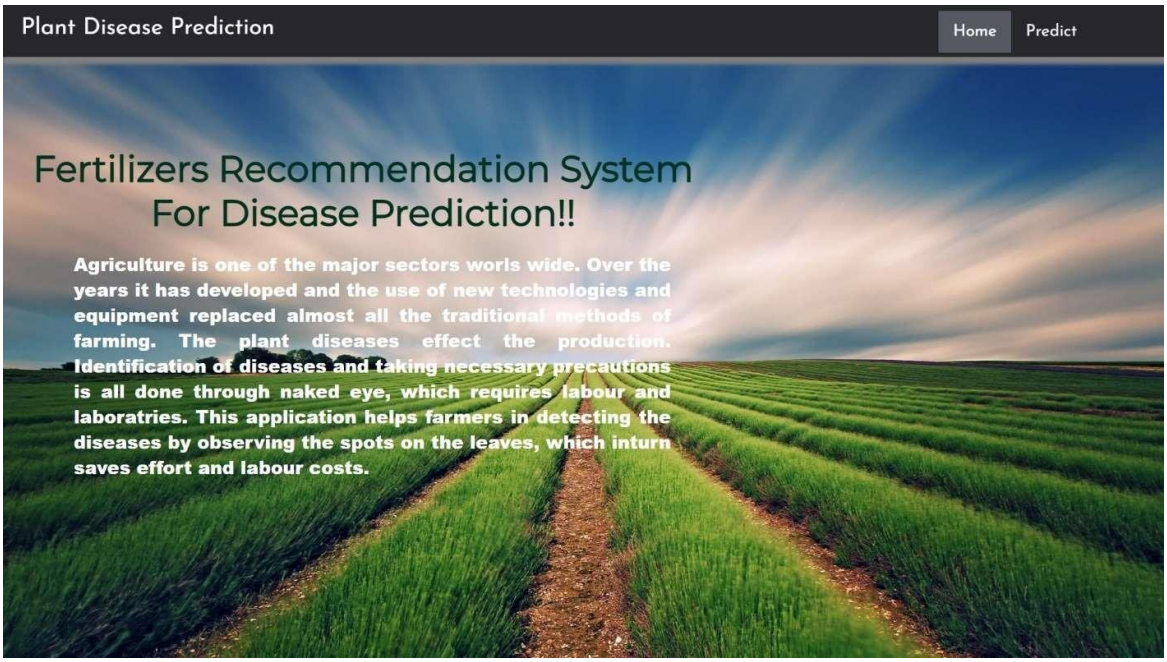
```
model.save('vegetables.h5')
```

CHAPTER-8

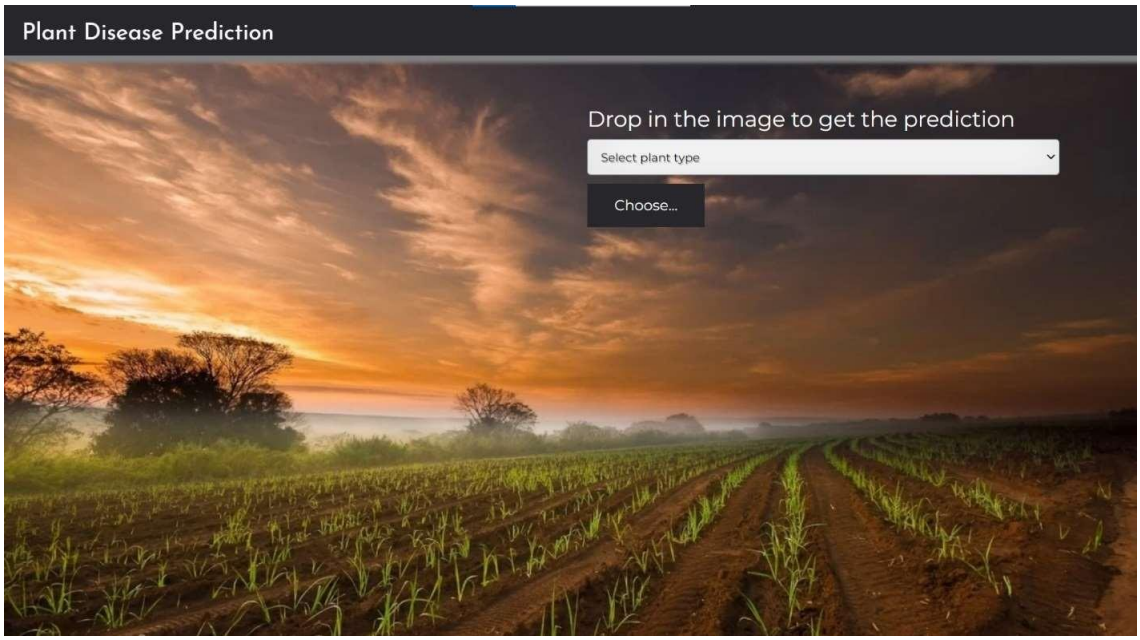
TESTING

8.1 TEST CASES

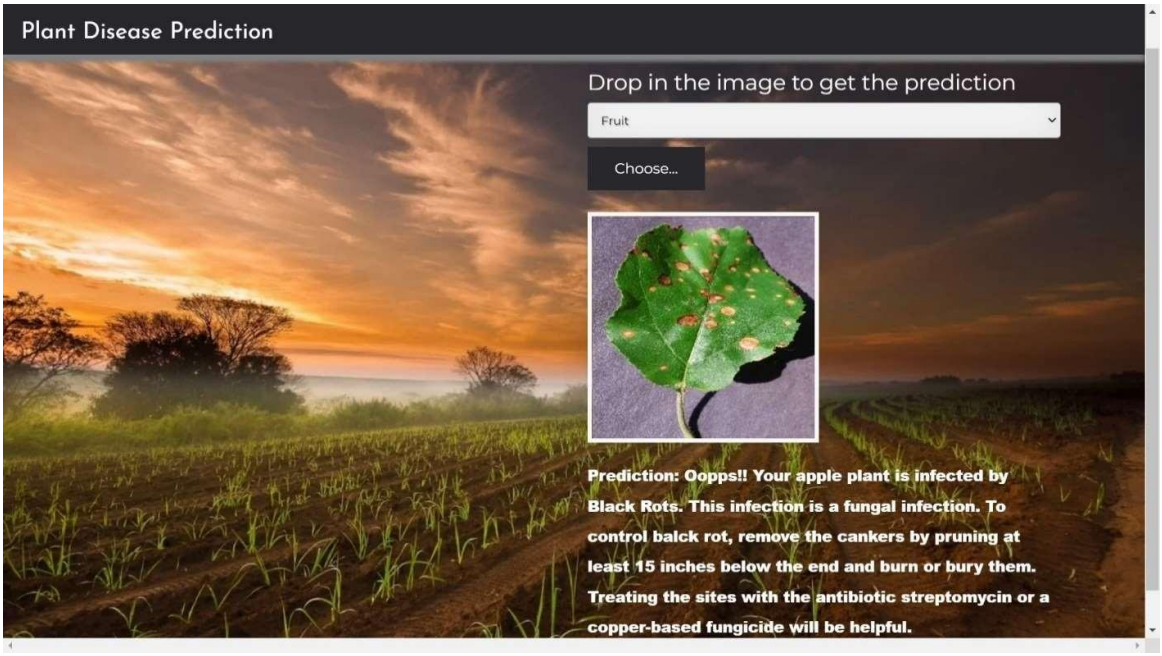
TEST CASE 1



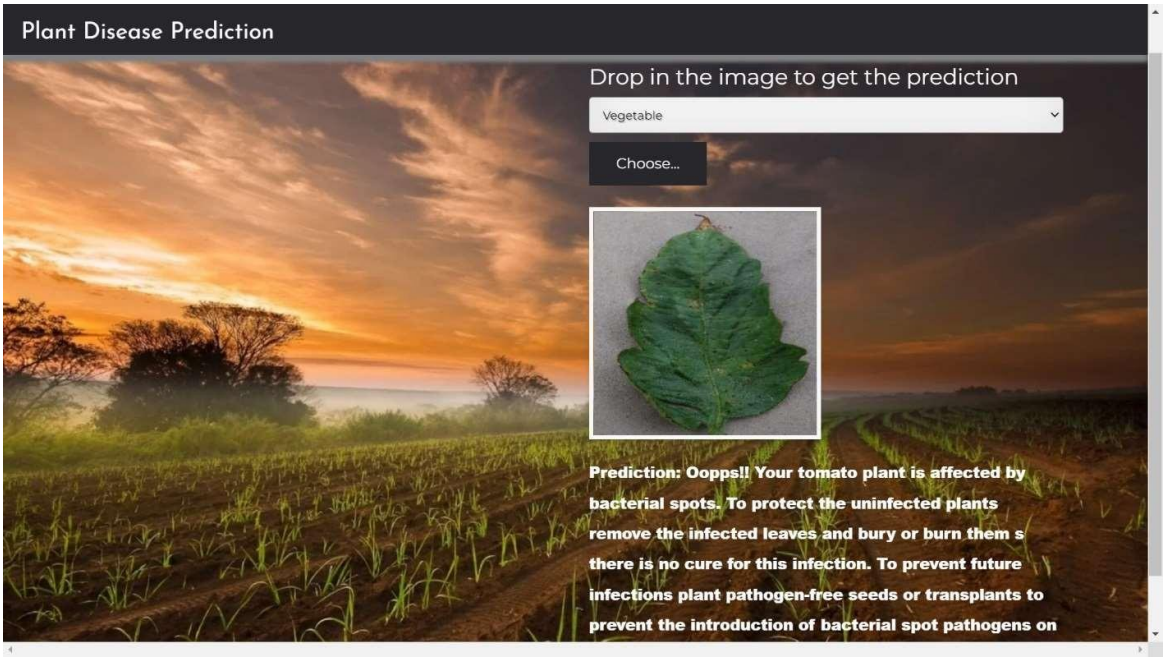
TEST CASE 2



TEST CASE 3



TEST CASE 4



8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Fertilizers Recommendation System for Disease Prediction] project at the time of the release to User Acceptance Testing (UAT).

2.Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Apple Black Rot	8	4	2	3	17
Northern Leaf Blight	9	5	3	6	23
Bacterial Spot	2	6	1	1	10
Early Blight	11	3	4	20	38
Late Blight	5	7	1	0	13
Leaf Mold	4	2	1	1	8
Septoria Leaf Spot	3	5	2	2	12
Totals	42	32	14	33	121

3.Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Apple Black Rot	17	0	0	17
Northern Leaf Blight	23	0	0	23
Bacterial Spot	10	0	0	10

Early Blight	38	0	0	38
Late Blight	13	0	0	13
Leaf Mold	8	0	0	8
Septoria Leaf Spot	12	0	0	12

CHAPTER-9
RESULTS

9.1 PERFORMANCE METRICS

9.1.1 Model Summary

The model summary table reports the strength of the relationship between the model and the dependent variable. R, the multiple correlation coefficient, is the linear correlation between the observed and model-predicted values of the dependent variable.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)            (None, 126, 126, 32)      896

 max_pooling2d_1 (MaxPooling  (None, 63, 63, 32)        0
 2D)

 flatten_1 (Flatten)          (None, 127008)            0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

9.1.2 Accuracy

It is one of the metrics to describe the accuracy of an algorithm on a classification task. Accuracy is the number of samples that are paired divided by the number of samples.

```
model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_te
st,validation_steps=len(x_test),epochs=10)

Epoch 1/10
171/171 [==============================] - 37s 215ms/step - loss: 0.6759 - ac
curacy: 0.8007 - val_loss: 0.3976 - val_accuracy: 0.8550
```

```
Epoch 2/10
171/171 [==============================] - 36s 212ms/step - loss: 0.2652 - ac
curacy: 0.9088 - val_loss: 0.2140 - val_accuracy: 0.9245
Epoch 3/10
171/171 [==============================] - 36s 210ms/step - loss: 0.2217 - ac
curacy: 0.9269 - val_loss: 0.2333 - val_accuracy: 0.9186
Epoch 4/10
171/171 [==============================] - 37s 218ms/step - loss: 0.1855 - ac
curacy: 0.9387 - val_loss: 0.2355 - val_accuracy: 0.9180
Epoch 5/10
171/171 [==============================] - 36s 210ms/step - loss: 0.1676 - ac
curacy: 0.9431 - val_loss: 0.1273 - val_accuracy: 0.9587
Epoch 6/10
171/171 [==============================] - 36s 212ms/step - loss: 0.1370 - ac
curacy: 0.9558 - val_loss: 0.1602 - val_accuracy: 0.9452
Epoch 7/10
171/171 [==============================] - 37s 218ms/step - loss: 0.1254 - ac
curacy: 0.9572 - val_loss: 0.1931 - val_accuracy: 0.9322
Epoch 8/10
171/171 [==============================] - 36s 211ms/step - loss: 0.1227 - ac
curacy: 0.9580 - val_loss: 0.1378 - val_accuracy: 0.9475
Epoch 9/10
171/171 [==============================] - 36s 210ms/step - loss: 0.1191 - ac
curacy: 0.9561 - val_loss: 0.1132 - val_accuracy: 0.9611
Epoch 10/10
171/171 [==============================] - 36s 210ms/step - loss: 0.1109 - ac
curacy: 0.9600 - val_loss: 0.1426 - val_accuracy: 0.9505
```

CHAPTER-10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- The main purpose of this system is to provide farmers with actionable fertilizer advice.
- Bridging information gaps through communication technology channels can overcome the problem of access to trained personnel, making knowledge more widely available.
- Not necessary to gain the knowledge about the fertilizers and disease.
- No need of physical observation to analyses the disease.
- The proposed model here produces very high accuracy of classification.

- Very High accuracy in image recognition problems.
- Automatically detects the important features without any human supervision.

10.2 DISADVANTAGES

- Lack of ability to be spatially invariant to the input data.
- Lots of training data is required.

CHAPTER-11
CONCLUSION

The Convolutional Neural Network model was implemented in TensorFlow backend system to classify the plant diseases. The same was implemented for real time data using Open CV. The most commonly used optimizers and activation functions were analyzed. In this report, we propose a user-friendly website based on deep learning called the 'Fertilizers Recommendation System for Disease Prediction'. With our system, we are successfully able to provide several features fertilizers recommendation and plant disease detection using CNN model on leaf images. The user can provide the input using forms on our user interface and quickly get their results.

CHAPTER-12
FUTURE SCOPE

This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy. For further betterment for the users, we will develop this website into an android app with some affordable features as we provide a separate portal for the purchasing of recommended fertilizers. The

user will get the scheduled notification about the usage and dosage of the fertilizers. Certain features will be added as the disease will be predicted even more accurately by submitting the soil test report from the user and also by the regional climatic conditions to which the plant grows.

CHAPTER-11
APPENDIX

Source Code

HTML

Home Page

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condense
d:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
```

```
<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin
Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header {
                top:0;
                margin:0px; left:
                0px; right: 0px;
                position: fixed;
                background-color: #28272c;
                color: white;
                box-shadow: 0px 8px 4px grey;
                overflow: hidden;
                padding-left:20px;
                font-family: 'Josefin Sans';
                font-size: 2vw;
                width: 100%;
                height:8%;
                text-align: center;
          }
          .topnav
  { overflow: hidden;
  background-color: #333;
}

.topnav-right a
  {float: left;
  color: #f2f2f2;

  text-align: center;
```

```
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover
  { background-color:
  #ddd;color: black;
}

.topnav-right a.active
  { background-color:
  #565961;color: white;
}

.topnav-right
  {float: right;
  padding-right:100px;
}

body {
  font-family:'Times New Roman', Times, serif;
  background-image: url("..static/images/s1.jpg");
  background-color:#ffffff;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
  .button {
  background-color: #28272c;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
```

```
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  border-radius: 12px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0
rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-
right:400px;}

input[type=text], input[type=password]
  {width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #28272c;
  color: white;
  padding: 14px 20px;
  margin-bottom:8px;
  border: none;
  cursor: pointer;
  width: 15%;
  border-radius:4px;
}

button:hover
  { opacity: 0.8;
```

```
  }

  .cancelbtn
    { width:
    auto;
    padding: 10px 18px;
    background-color:  #f44336;
  }

  .imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
  }

  img.avatar
    { width: 30%;
    border-radius: 50%;
  }

  .container
    { padding: 16px;
  }

  span.psw
    { float:
    right;
    padding-top: 16px;
  }
  /* Change styles for span and cancel button on extra small
screens */
  @media screen and (max-width: 300px)
    {span.psw {
       display: block;
       float: none;}
```

```
.cancel btn { width:
100%;
    }
  }

  .home{
       margin:80px;

    width:   84%;
    height: 500px;
    padding-top:10px;
    padding-left: 30px;

  }
  .login{
       margin:80px;
       box-sizing: content-box;
    width: 84%;
    height: 420px;
    padding: 30px;
    border: 10px solid blue;
  }
  .left,.right{
   box-sizing: content-box;
   height: 400px;
   margin:20px;
   border: 10px solid blue;
  }

  .mySlides {display:  none;}
  img {vertical-align: middle;}

  /* Slideshow container */
  .slideshow-container {
```

```
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom:   8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active {
  background-color: #717171;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
```

```
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade
  {from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade
  { from
  {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
</style>
</head>

<body>

<div class="header">
 <div  style="width:50%;float:left;font-size:2vw;text-
align:left;color:white;  padding-top:1%">Plant  Disease
Prediction</div>
  <div  class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('prediction')}}">Predict</a>
  </div>
</div>
```

```
<div  style="background-color:#ffffff;">
<div  style="width:60%;float:left;">
<div style="font-size:40px;color:#013220;font-
family:Montserrat;padding-left:20px;text-align:center;padding-
top:10%;">
<b>Fertilizers Recommendation System<br> For Disease
Prediction!!</b>
</div><br>
<div style="font-size:20px;color:#ffffff;font-family:Arial
Black;padding-left:70px;padding-right:30px;text-
align:justify;">Agriculture is one of the major sectors worls
wide. Over the years it has developed and the use of new
technologies and equipment replaced almost all the traditional
methods of farming. The plant diseases effect the production.
Identification of diseases and taking necessary precautions is
all done through naked eye, which requires labour and
laboratries. This application helps farmers in detecting the
diseases by observing the spots on the leaves, which inturn
saves effort and labour costs.</div><br><br>
</div>
</div>
<div  style="width:40%;float:right;"><br><br>
</div>
</div>

<div class="home">
<br>
</div>

<script>
var slideIndex = 0;
showSlides();

function showSlides() {
```

```
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++)
    {slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active",
"");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

Predict Page

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
```

```
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.
css" rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"
></script>
    <script
src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></scrip
t>
    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js
"></script>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condense
d:300' rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin
Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<link href="{{ url_for('static', filename='css/final.css') }}"
rel="stylesheet">
<style>
.header {
                top:0;
                margin:0px; left:
                0px; right: 0px;
                position: fixed;
                background-color: #28272c;
```

```
                color: white;
                box-shadow: 0px 8px 4px grey;
                overflow: hidden;
                padding-left:20px;
                font-family: 'Josefin Sans';
                font-size: 2vw;
                width: 100%;
                height:8%;
                text-align: center;
            }
            .topnav
  { overflow: hidden;
  background-color: #333;
}

.topnav-right a
  { float:   left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover
  { background-color:
  #ddd;color: black;
}

.topnav-right a.active
  { background-color:
  #565961;color: white;
}
```

```
  .topnav-right
    {float: right;
    padding-right:100px;
  }

  .login{
  margin-top:-70px;
  }
  body {

   background-image: url("../static/images/s2.jpg");
    background-color:#ffffff;
    background-repeat: no-repeat;
    background-size:cover;
    background-position: 0px 0px;
    }
  .login{
       margin-top:100px;
  }
.container {
 margin-top:40px;
 padding: 16px;
  }
  select {
       width: 100%;
       margin-bottom: 10px;
       background:  rgba(255,255,255,255);
       border: none;
       outline: none;
       padding: 10px;
       font-size: 13px;
       color: #000000;
       text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
```

```
     border: 1px solid rgba(0,0,0,0.3);
     border-radius: 4px;
     box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px
1px rgba(255,255,255,0.2);
     -webkit-transition: box-shadow .5s ease;
     -moz-transition: box-shadow .5s ease;
     -o-transition: box-shadow .5s ease;
     -ms-transition: box-shadow .5s ease;
     transition: box-shadow .5s ease;
}


</style>
</head>

<body  style="font-family:Montserrat;overflow:scroll;">

<div class="header">
 <div  style="width:50%;float:left;font-size:2vw;text-
align:left;color:white;  padding-top:1%">Plant  Disease
Prediction</div>
  <div  class="topnav-right"  style="padding-top:0.5%;">


  </div>
</div>
<div  class="container">
        <div  id="content"  style="margin-top:2em">
          <div  class="container">
            <div  class="row">
               <div  class="col-sm-6 bd" >

                 <br>
               </div>
```

```
                <div class="col-sm-6">
                     <div>
                          <h4>Drop in the image to get the
prediction  </h4>
                <form action = "/precations/precations_veg"
id="upload-file"  method="post"  enctype="multipart/form-data">
                     <select  name="plant">

                           <option  value="select"
selected>Select plant type</option>
                           <option  value="fruit">Fruit</option>
                           <option
value="vegetable">Vegetable</option>
          </select><br>
                     <label  for="imageUpload"  class="upload-
label" style="background: #28272c;">
                          Choose...
                     </label>
                     <input  type="file"  name="image"
id="imageUpload" accept=".png, .jpg, .jpeg">
                </form>


                <div  class="image-section"  style="display:none;">
                     <div  class="img-preview">
                          <div  id="imagePreview">
                          </div>
                     </div>
                     <div>
                          <button type="button" class="btn btn-
info btn-lg " id="btn-predict" style="background:
#28272c;">Predict!</button>
                     </div>
                </div>
```

```
                <div class="loader" style="display:none;"></div>

                <h3>
                     <span id="result" style="font-
size:17px;color:white;font-family:Arial Black;"> </span>
                </h3>

          </div>
                </div>

            </div>
          </div>
          </div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>
</html>
```

CSS

```
.img-preview
    { width: 256px;
    height: 256px;
    position:  relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}
```

```
.img-preview>div
    {width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"]
    {display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #28272c;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-
    label:hover{ backgroun
    d: #C2C5A8;color:
    #39D2B4;
}
.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #28272c; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
```

```
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

Java Script

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload  Preview
    function readURL(input) {
        if (input.files && input.files[0])
            {var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image',
'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
        readURL(this);
    });
```

```
    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text('Prediction:  '+data);
                console.log('Success!');
            },
        });
    });

});
```

App.py

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
```

```
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect,
url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session
app = Flask(__name__)
global sess
global graph
graph=tf.compat.v1.get_default_graph()
model = load_model(r"C:\Users\Admin\Desktop\New
folder\vegetables.h5")
model1=load_model(r"C:\Users\Admin\Desktop\New folder\fruit.h5")
@app.route('/')
def home():
    return  render_template('home.html')
@app.route('/prediction')
def prediction():
    return  render_template('predict.html')
@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':

        f = request.files['image']
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(128, 128))

        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
```

```
        plant=request.form['plant']
        print(plant)

        if(plant=="vegetable"):
            preds = model.predict(x)
            preds = np.argmax(preds)
            print(preds)
            df=pd.read_excel('precautions_veg.xlsx')
            print(df.iloc[preds]['caution'])

        else:
            preds = model1.predict(x)
            preds = np.argmax(preds)

            df=pd.read_excel('precautions_fruits.xlsx')
            print(df.iloc[preds]['caution'])



        return  df.iloc[preds]['caution']

if___name__ == "__main__":
    app.run(debug=True)
```

Python Code

Fruit

```
from tensorflow.keras.preprocessing.image import
ImageDataGenerator

train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip
=True)
test_datagen = ImageDataGenerator(rescale=1./255,)

x_train =
train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/D
ataset/Dataset Plant Disease/fruit-dataset/fruit-
```

```
dataset/train',target_size = (128,128),batch_size = 32,
class_mode = 'categorical')
Found 5448 images belonging to 6 classes.

x_test =
test_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/Da
taset/Dataset Plant Disease/fruit-dataset/fruit-
dataset/test',target_size = (128,128),batch_size = 32,class_mode
= 'categorical')
Found 1696 images belonging to 6 classes.

x_train.class_indices

{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import
Dense,Convolution2D,MaxPooling2D,Flatten

model=Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activat
ion='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.summary()
Model: "sequential"
_________________________________________________________________
_
 Layer (type)                Output Shape              Param #
=================================================================
=
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

=================================================================
=
Total params: 896
Trainable params: 896
Non-trainable params: 0
```

```
_____________________________________________________________________
_

model.add(Dense(6,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adam',m
etrics=['accuracy'])

len(x_train)

171
model.fit_generator(x_train,steps_per_epoch=len(x_train),validat
ion_data=x_test,validation_steps=len(x_test),epochs=10)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
UserWarning: `Model.fit_generator` is deprecated and will be rem
oved in a future version. Please use `Model.fit`, which supports
generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
171/171 [==============================] - 1310s 8s/step - loss:
1.1538 - accuracy: 0.7673 - val_loss: 0.2983 - val_accuracy: 0.8
856
Epoch 2/10
171/171 [==============================] - 37s 214ms/step - loss
: 0.2908 - accuracy: 0.9005 - val_loss: 0.2164 - val_accuracy: 0
.9204
Epoch 3/10
171/171 [==============================] - 36s 211ms/step - loss
: 0.2305 - accuracy: 0.9174 - val_loss: 0.2043 - val_accuracy: 0
.9334
Epoch 4/10
171/171 [==============================] - 36s 212ms/step - loss
: 0.2040 - accuracy: 0.9326 - val_loss: 0.1546 - val_accuracy: 0
.9422
Epoch 5/10
171/171 [==============================] - 36s 213ms/step - loss
: 0.1913 - accuracy: 0.9356 - val_loss: 0.1513 - val_accuracy: 0
.9446
Epoch 6/10
171/171 [==============================] - 36s 211ms/step - loss
: 0.1677 - accuracy: 0.9425 - val_loss: 0.2398 - val_accuracy: 0
.9186
Epoch 7/10
171/171 [==============================] - 36s 209ms/step - loss
: 0.1564 - accuracy: 0.9449 - val_loss: 0.1159 - val_accuracy: 0
.9552
Epoch 8/10
171/171 [==============================] - 36s 211ms/step - loss
: 0.1451 - accuracy: 0.9521 - val_loss: 0.1630 - val_accuracy: 0
.9475
Epoch 9/10
```

```
171/171 [==============================] - 36s 213ms/step - loss
: 0.1404 - accuracy: 0.9514 - val_loss: 0.1704 - val_accuracy: 0
.9404
Epoch 10/10
171/171 [==============================] - 36s 211ms/step - loss
: 0.1150 - accuracy: 0.9598 - val_loss: 0.1693 - val_accuracy: 0
.9452

model.save('fruit.h5')
```

Vegetable

```
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip
=True)
test_datagen = ImageDataGenerator(rescale=1./255,)
x_train =
train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/D
ataset/Dataset Plant Disease/Veg-dataset/Veg-
dataset/train_set',target_size = (128,128),batch_size = 32,
class_mode = 'categorical')
Found 11408 images belonging to 9 classes.
x_test =
test_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/Da
taset/Dataset Plant Disease/Veg-dataset/Veg-
dataset/test_set',target_size = (128,128),batch_size =
32,class_mode = 'categorical')
Found 3436 images belonging to 9 classes.
x_train.class_indices
{'Pepper,_bell___Bacterial_spot': 0,
 'Pepper,_bell___healthy': 1,
 'Potato___Early_blight': 2,
 'Potato___Late_blight': 3,
 'Potato___healthy': 4,
 'Tomato___Bacterial_spot': 5,
 'Tomato___Late_blight': 6,
 'Tomato___Leaf_Mold': 7,
 'Tomato___Septoria_leaf_spot': 8}
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import
Dense,Convolution2D,MaxPooling2D,Flatten
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activat
ion='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary() Model:
"sequential_1"
_________________________________________________________________
_
```

```
 Layer (type)                 Output Shape              Param #
=================================================================
=
 conv2d_1 (Conv2D)            (None, 126, 126, 32)      896

 max_pooling2d_1 (MaxPooling  (None, 63, 63, 32)        0
 2D)

 flatten_1 (Flatten)          (None, 127008)            0

=================================================================
=
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
_
model.add(Dense(9,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',m
etrics=['accuracy'])
len(x_train)
357
model.fit_generator(x_train,steps_per_epoch=len(x_train),validat
ion_data=x_test,validation_steps=len(x_test),epochs=10)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
UserWarning: `Model.fit_generator` is deprecated and will be
removed in a future version. Please use `Model.fit`, which
supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
357/357 [==============================] - 3005s 8s/step - loss:
2.1432 - accuracy: 0.1668 - val_loss: 2.1199 - val_accuracy:
0.1772
Epoch 2/10
357/357 [==============================] - 77s 216ms/step -
loss: 2.0963 - accuracy: 0.1698 - val_loss: 2.0906 -
val_accuracy: 0.1941
Epoch 3/10
357/357 [==============================] - 76s 213ms/step -
loss: 2.0807 - accuracy: 0.1864 - val_loss: 2.0851 -
val_accuracy: 0.1941
Epoch 4/10
357/357 [==============================] - 77s 215ms/step -
loss: 2.0764 - accuracy: 0.1864 - val_loss: 2.0833 -
val_accuracy: 0.1941
Epoch 5/10
357/357 [==============================] - 76s 213ms/step -
loss: 2.0748 - accuracy: 0.1864 - val_loss: 2.0825 -
val_accuracy: 0.1941
Epoch 6/10
```

```
357/357 [==============================] - 77s 216ms/step -
loss: 2.0740 - accuracy: 0.1864 - val_loss: 2.0822 -
val_accuracy: 0.1941
Epoch 7/10
357/357 [==============================] - 76s 212ms/step -
loss: 2.0737 - accuracy: 0.1864 - val_loss: 2.0822 -
val_accuracy: 0.1941
Epoch 8/10
357/357 [==============================] - 76s 212ms/step -
loss: 2.0735 - accuracy: 0.1864 - val_loss: 2.0819 -
val_accuracy: 0.1941
Epoch 9/10
357/357 [==============================] - 77s 215ms/step -
loss: 2.0736 - accuracy: 0.1864 - val_loss: 2.0819 -
val_accuracy: 0.1941
Epoch 10/10
357/357 [==============================] - 76s 213ms/step -
loss: 2.0736 - accuracy: 0.1864 - val_loss: 2.0815 -
val_accuracy: 0.1941
model.save('vegetables.h5')
```

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-18625-1659687882

PROJECT DEMO LINK

https://drive.google.com/drive/folders/1hqqQAOSiJ3forddvx6MPJalbpGLqLbhR